Subject Name Digital Image Processing
Code : MTCS-202 (C)
M.Tech (Computer Science)
2nd Semester
Faculty Name: Dr Satya Bhushan Verma

Edge detection and line detection in Image processing

Edges: Edges are abrupt changes in intensity, discontinuity in image brightness or contrast; usually edges occur on the boundary of two regions.

Figure: Original image (left) and edge (right)

Edge detection: Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision.

Why we use edge detection?

- Reduce unnecessary information in the image while preserving the structure of the image.
- Extract important features of an image such as corners, lines, and curves.
- Edges provide strong visual clues that can help the recognition process.

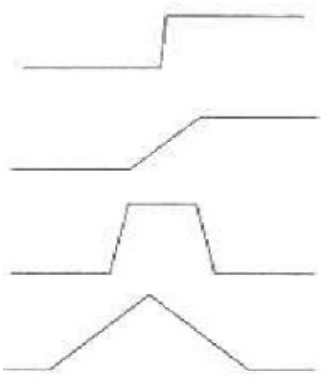
Type of edges

1: Step Type

2: Ramp type

3: Ridges

4: Roof



Here are some of the masks for edge detection.

- Prewitt Operator
- Sobel Operator
- Robinson Compass Masks
- Kirsch Compass Masks
- Laplacian Operator.

Prewitt Operator: Prewitt operator is used for edge detection in an image. By using Prewitt operator we can detects only horizontal and vertical edges.

For vertical Edges

Mask for detection of vertical edges

-1	0	1
-1	0	1
-1	0	1

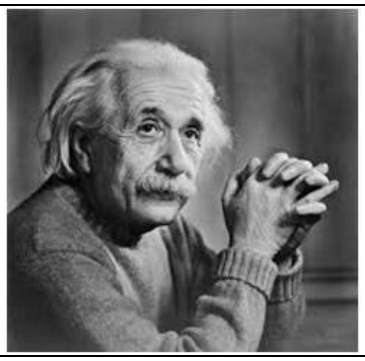
Above mask will find the edges in vertical direction and it is because the zeros column in the vertical direction. When you will convolve this mask on an image, it will give you the vertical edges in an image.

Horizontal Edges direction

Mask for detection of horizontal edges

-1	-1	-1
0	0	0
1	1	1

Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction. When you will convolve this mask onto an image it would prominent horizontal edges in the image.

		
Original Image	Applying Horizontal Mask	Applying Horizontal Mask

Sobel Operator: The sobel operator is very similar to Prewitt operator. It is also a derivate mask and is used for edge detection. Like Prewitt operator sobel operator is also used to detect two kinds of edges in an image:

- Vertical direction
- Horizontal direction

How to differ from Prewitt Operator

The main difference is that in Sobel operator the coefficients of masks are not fixed and they can be adjusted according to our requirement unless they do not violate any property of derivative masks.

For vertical Edges

Vertical Mask of Sobel Operator

-1	0	1
-2	0	2
-1	0	1

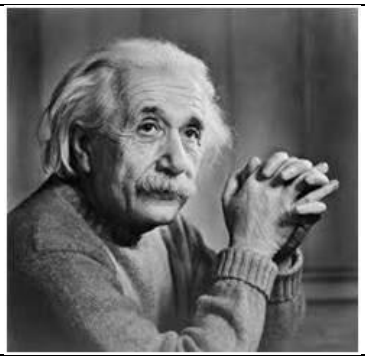
This mask works exactly same as the Prewitt operator vertical mask. There is only one difference that is it has “2” and “-2” values in center of first and third column. When applied on an image this mask will highlight the vertical edges.

Horizontal Edges direction

Horizontal Mask of Sobel Operator

-1	-2	-1
0	0	0
1	2	1

Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction. When you will convolve this mask onto an image it would prominent horizontal edges in the image. The only difference between it is that it have 2 and -2 as a centre element of first and third row.

		
Original Image	Applying Horizontal Mask	Applying Horizontal Mask

As you can see that in the first picture on which we apply vertical mask, all the vertical edges are more visible than the original image. Similarly in the second picture we have applied the horizontal mask and in result all the horizontal edges are visible.

So in this way you can see that we can detect both horizontal and vertical edges from an image. Also if you compare the result of sobel operator with Prewitt operator, you will find that sobel operator finds more edges or make edges more visible as compared to Prewitt Operator. This is because in sobel operator we have allotted more weight to the pixel intensities around the edges.

Applying more weight to mask

Now we can also see that if we apply more weight to the mask, the more edges it will get for us. Also as mentioned in the start of the tutorial that there is no fixed coefficients in sobel operator, so here is another weighted operator

-1	0	1
-5	0	5
-1	0	1

Vertical Mask

-1	-5	-1
0	0	0
1	5	1

Horizontal Mask

Robinson Compass Masks: A Robinson compass mask is a type of mask which is used for edge detection. It has eight orientations. It is also known as the direction mask. It extracts the edges with respect to its direction. Following are its eight orientations:

- North,
- North West
- West,
- South West
- South,
- South East
- East,
- North East

There is no fixed mask. You can take any mask and you have to rotate it to find edges in all the above mentioned directions. All the masks are rotated on the bases of direction of zero columns.

0	1	2
-1	0	1
-2	-1	0

North West Mask

-1	0	1
-2	0	2
-1	0	1

North Mask

-2	-1	0
-1	0	1
0	1	2

North East Mask

1	2	1
0	0	0
-1	-2	-1

West Mask

Robinson Compass Masks

-1	-2	-1
0	0	0
1	2	1

East Mask

2	1	0
1	0	-1
0	-1	-2

South West Mask

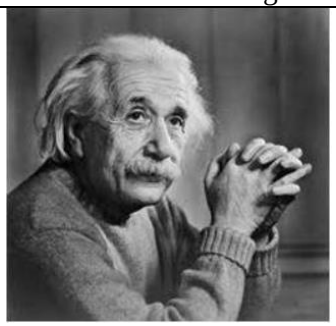
1	0	-1
2	0	-2
1	0	-1

South Mask

0	-1	-2
1	0	-1
2	1	0

South East Mask

As you can see that all the directions are covered on the basis of zeros direction. Each mask will give you the edges on its direction. Now let's see the result of the entire above masks. Suppose we have a sample picture from which we have to find all the edges. Here is our sample picture:

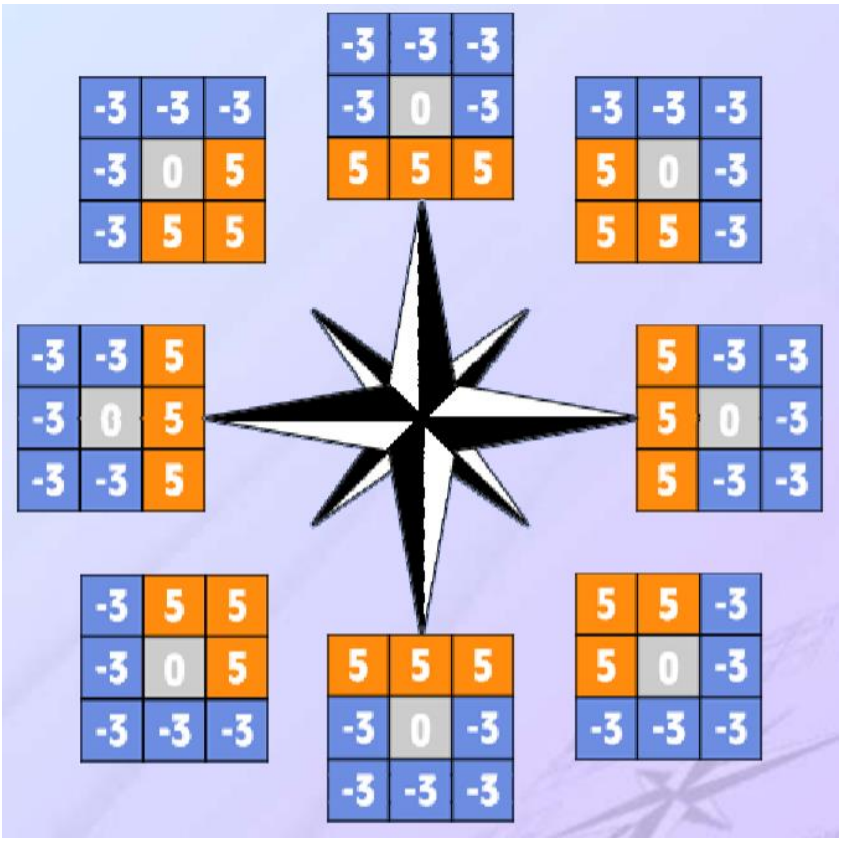
 North West Direction Edges	 North Direction Edges	 North East Direction Edges
 West Direction Edges	 Original Image	 East Direction Edges
 South West Direction Edges	 South Direction Edges	 South East Direction Edges

Kirsch Compass Masks: The Kirsch operator or Kirsch compass kernel is a non-linear edge detector that finds the maximum edge strength in a few predetermined directions. It is named after the computer scientist Russell A. Kirsch.

This is also like Robinson compass find edges in all the eight directions of a compass. The only difference between Robinson and kirsch compass masks is that in Kirsch we have a standard mask but in Kirsch we change the mask according to our own requirements.

With the help of Kirsch Compass Masks we can find edges in the following eight directions.

- North
- North West
- West
- South West
- South
- South East
- East
- North East



8 different Kirsch Compass Masks

Laplacian Operator: Laplacian Operator is also a derivative operator which is used to find edges in an image. The major difference between Laplacian and other operators like Prewitt, Sobel, Robinson and Kirsch is that these all are first order derivative masks but Laplacian is a second order derivative mask. In this mask we have two further classifications one is Positive Laplacian Operator and other is Negative Laplacian Operator.

Laplacian is a derivative operator; its uses highlight gray level discontinuities in an image and try to deemphasize regions with slowly varying gray levels. This operation in result produces such images which have grayish edge lines and other discontinuities on a dark background. This produces inward and outward edges in an image

0	1	0
1	-4	1
0	1	0

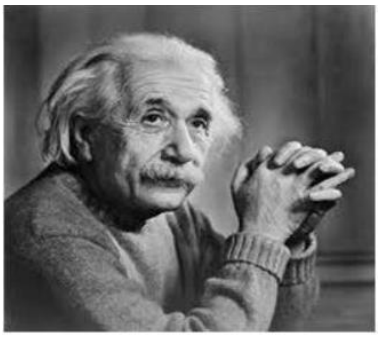
Positive Laplacian Operator

0	-1	0
-1	4	-1
0	-1	0

Negative Laplacian Operator

The important thing is how to apply these filters onto image. Remember we can't apply both the positive and negative Laplacian operator on the same image. we have to apply just one but the thing to remember is that if we apply positive Laplacian operator on the image then we subtract the resultant image from the original image to get the sharpened image. Similarly if we apply negative Laplacian operator then we have to add the resultant image onto original image to get the sharpened image.

Let's apply these filters onto an image and see how it will get us inward and outward edges from an image. Suppose we have a following sample image.



Original Image

Image After applying positive Laplacian operator

Image After applying Negative Laplacian operator

Line Detection: In image processing, line detection is an algorithm that takes a collection of n edge points and finds all the lines on which these edge points lie. The most popular line detectors are the Hough transform and convolution-based techniques.

Hough transform: The Hough transform is an incredible tool that lets you identify lines. Not just lines, but other shapes as well.

Example: Using Hough transform show that the points (1,1), (2,2), and (3, 3) are collinear find the equation of line.

Solution:

The equation of line is y=mx+c

In order to perform Hough transform we need to convert line from (x,y) plane to (m,c) plane

Equation of (m,c) plane is

Step 1:

C=mx+y

For (1,1)

y=mx+c

1=m+c

C=-m+1

If c=0 then (0=-m+1) m=1

If m=0 then (c=1) c=1

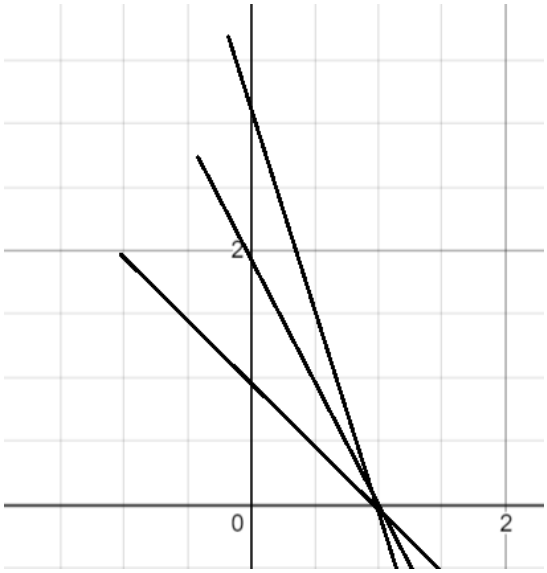
(m,c) = (1, 1)

Similarly for

If (x,y)= (2,2) then (m,c) = (1,2)
If (x,y)= (3,3) then (m,c) = (1,3)

Step 2:

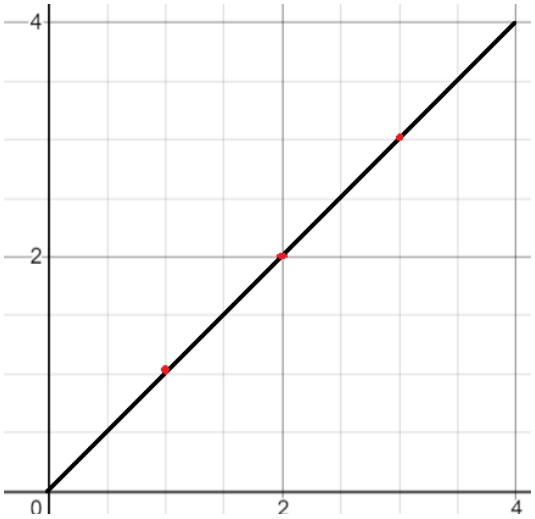
Plot a graph for (mc) = (1, 1), (1,2), and (1,3)



Intersect at the point (0,1)
Then (m,c) = (0,1)

Step 3 : The original equation of line is (y=mx+c) put the value of m and **c** on this eqe.
Then **y=x**

points (1,1), (2,2), and (3, 3) are collinear



Convolution-Based Techniques: The line detection operator consists of a convolution kernel tuned to detect the presence of lines of a particular width n, at a particular orientation. Figure 1 shows a collection of four such kernels, which each respond to lines of single pixel width at the particular orientation shown.

a)

−1	−1	−1
2	2	2
−1	−1	−1

b)

−1	2	−1
−1	2	−1
−1	2	−1

c)

−1	−1	2
−1	2	−1
2	−1	−1

d)

2	−1	−1
−1	2	−1
−1	−1	2

Figure 1 Four line detection kernels which respond maximally to horizontal, vertical and oblique (+45 and - 45 degree) single pixel wide lines.

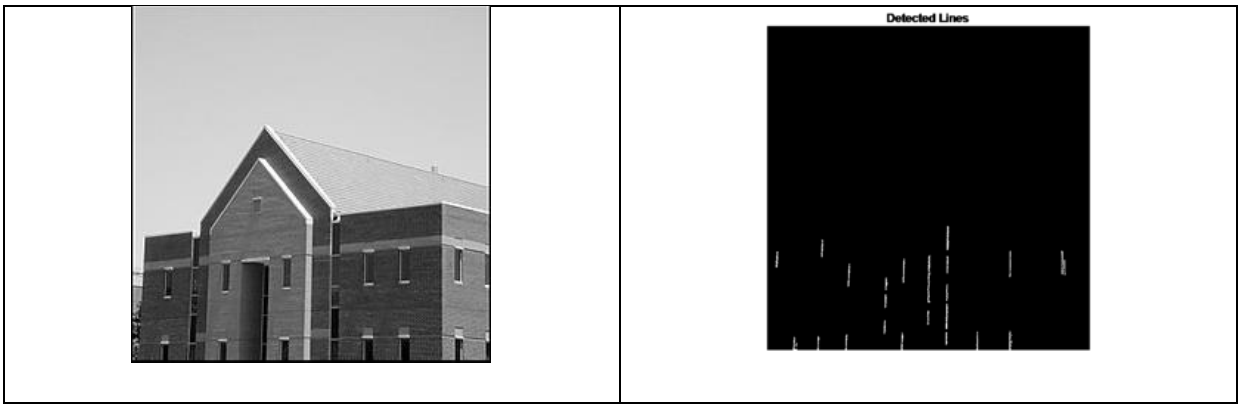


Figure: Original image and detection of lines (only vertical lines)